adjBaseOlig.error  Evaluates LPE variance function of M for quantiles of A within and experimental condition and then interpolates it for all genes.

Description
Calls adjBaseOlig.error.step1 and adjBaseOlig.error.step2 functions in order to calculate the baseline distribution.

Usage
adjBaseOlig.error(y, stats=median, q=0.01, min.genes.int=10, div.factor=1, setMax1=FALSE)

Arguments
y  y is a preprocessed matrix or data frame of expression intensities in which columns are expression intensities for a particular experimental condition and rows are genes.

stats  It determines whether mean or median is to be used for the replicates

q  q is the quantile width; q=0.01 corresponds to 100 quantiles i.e. percentiles. Bins/quantiles have equal number of genes and are split according to the average intensity A.

min.genes.int  Determines the minimum number of genes in a subinterval for selecting the adaptive intervals.

div.factor  Determines the factor by which sigma needs to be divided for selecting adaptive intervals.

setMax1  If T then all variances below the max variance in the ordered distribution of variances are set to the maximum variance. If F then variances are left as is (recommended)

Value
Returns object of class baseOlig comprising a data frame with 2 columns: A and var M, and rows for each quantile specified. The A column contains the median values of A for each quantile/bin and the M columns contains the pooled variance of the replicate chips for genes within each quantile/bin.
adjBaseOlig.error.step1

Evaluates LPE variance function of M for quantiles of A within and experimental condition by dividing the A in 100 intervals.
Description

Genes are placed in bins/quantiles according to their average expression intensity. The function adjBaseOlig.error calculates a pooled variance of M for genes within these bins/quantiles of A for the replicates of the experimental condition contained in y. Here the assumption is that variance of the genes in each interval is similar.

Usage

adjBaseOlig.error.step1(y, stats=median, setMax=FALSE, q=0.01, df=10)

Arguments

y

y is a preprocessed matrix or data frame of expression intensities in which columns are expression intensities for a particular experimental condition and rows are genes.

stats

It determines whether mean or median is to be used for the replicates

setMax

If T then all variances below the max variance in the ordered distribution of variances are set to the maximum variance. If F then variances are left as is (recommended)

q

q is the quantile width; q=0.01 corresponds to 100 quantiles i.e. percentiles. Bins/quantiles have equal number of genes and are split according to the average intensity A.

df

df stands for degrees of freedom. It is used in smooth.spline function to interpolate the variances of all genes. Default value is 10.

Value

Returns object of class baseOlig, comprising a data frame with 2 columns: A and var M. The A column contains the median values of each gene and the M columns contains the corresponding variance. Number of rows of the data-frame is same as that of the number of genes.

Author(s)

Carl Murie (carl.murie@mcgill.ca), Nitin Jain (nitin.jain@pfizer.com)

References


Jain et. al. (2003) Local pooled error test for identifying differentially expressed genes with a small number of replicated microarrays, Bioinformatics, 1945-1951.


See Also

lpeAdj
Examples

```r
# Loading the library and the data
library(LPEadj)
data(Ley)
dim(Ley)
# Gives 12488 by 7
Ley[,]
# Returns
# ID c1 c2 c3 t1 t2 t3
# 1 AFFX-MurIL2_at 4.06 3.82 4.28 11.47 11.54 11.34
# 2 AFFX-MurIL10_at 4.56 2.79 4.83 4.25 3.72 2.94
# 3 AFFX-MurIL4_at 5.14 4.10 4.59 4.67 4.71 4.67

Ley[,2:7] <- preprocess(Ley[,2:7], data.type="MAS5")
# Finding the baseline distribution of subset of the data
# condition one (3 replicates)
var.1 <- adjBaseOlig.error.step1(Ley[,2:4], setMax=FALSE, q=0.01)
dim(var.1)
# Returns a matrix of 1000 by 2 \( (A,M) \) format
```

adjBaseOlig.error.step2

Evaluates LPE variance function of M for quantiles of A within and in experimental condition. It is based on the adaptive number of intervals.

Description

Similar to adjBaseOlig.error.step1 function, except that now the number of bins are chosen adaptively instead of fixed 100.

Usage

adjBaseOlig.error.step2(y, baseOlig.error.step1.res, df=10, stats=median, setMax=FALSE)

Arguments

- **y**: y is a preprocessed matrix or data frame of expression intensities in which columns are expression intensities for a particular experimental condition and rows are genes.
- **baseOlig.error.step1.res**: It is the result obtained from adjBaseOlig.error.step1 function, in which number of bins are fixed=100.
- **df**: df stands for degrees of freedom. It is used in smooth.spline function to interpolate the variances of all genes. Default value is 10.
- **stats**: It determines whether mean or median is to be used for the replicates.
- **setMax**: If T then all variances below the max variance in the ordered distribution of variances are set to the maximum variance. If F then variances are left as is (recommended).
min.genes.int
Determine the minimum number of genes in a subinterval for selecting the adaptive intervals.

div.factor
Determines the factor by which sigma needs to be divided for selecting adaptive intervals.

Value
Returns object of class baseOlig comprising a data frame with 2 columns: A and var M, and rows for each quantile specified. The A column contains the median values of A for each quantile/bin and the M columns contains the pooled variance of the replicate chips for genes within each quantile/bin.

Author(s)
Carl Murie (carl.murie@mcgill.ca), Nitin Jain (nitin.jain@pfizer.com)

References

Jain et. al. (2003) _Local pooled error test for identifying differentially expressed genes with a small number of replicated microarrays_, Bioinformatics, 1945-1951.


See Also
lpeAdj

Examples

```r
# Loading the library and the data
library(LPEadj)
data(Ley)
dim(Ley)
# Gives 12488 by 7
Ley[1:3,]
# Returns
# ID c1 c2 c3 t1 t2 t3
# 1 AFFX-MurIL2_at 4.06 3.82 4.28 11.47 11.54 11.34
# 2 AFFX-MurIL10_at 4.56 2.79 4.83 4.25 3.72 2.94
# 3 AFFX-MurIL4_at 5.14 4.10 4.59 4.67 4.71 4.67

Ley[1:1000,2:7] <- preprocess(Ley[1:1000,2:7],data.type="MAS5")
# Finding the baseline distribution of subset of the data
# condition one (3 replicates)
var.1 <- adjBaseOlig.error.step1(Ley[1:1000,2:4], q=0.01, df=10)
dim(var.1)
var.11 <- adjBaseOlig.error.step2(Ley[1:1000,2:4], var.1, df=10)
# Returns a matrix of 1000 by 2 (A,M) format
```
calculateLpeAdj  Evaluates local pooled error significance test with user chosen variance adjustments.

Description

The local pooled error test attempts to reduce dependence on the within-gene estimates in tests for differential expression, by pooling error estimates within regions of similar intensity. Note that with the large number of genes there will be genes with low within-gene error estimates by chance, so that some signal-to-noise ratios will be large regardless of mean expression intensities and fold-change. The local pooled error attempts to avert this by combining within-gene error estimates with those of genes with similar expression intensity.

Usage

```r
calculateLpeAdj(x, y, basevar.x, basevar.y, df=10, array.type="olig", basevar.y, df=10, array.type="olig", probe.set.name="OLIG.probe.name", trim.percent=5, adjust1=1.57, adjust2=1.57)
```

Arguments

- `x`: Replicated data from first experimental condition (as matrix or data-frame)
- `y`: Replicated data from second experimental condition (as matrix or data-frame)
- `basevar.x`: Baseline distribution of first condition obtained from function baseOlig.error
- `basevar.y`: Baseline distribution of second condition obtained from function baseOlig.error
- `df`: Degrees of freedom used in fitting smooth.spline to estimates of var.M for bins in A
- `array.type`: Currently supports oligo arrays
- `probe.set.name`: Gene IDs. By default if they are not provided then 1,2,3,... is assigned as GeneID
- `trim.percent`: Percent of \((A, \text{var.M})\) estimates to trim from low end of A
- `adjust1`: adjustment factor of variance for first group
- `adjust2`: adjustment factor of variance for second group

Details

The LPE test statistic numerator is the difference in medians between the two experimental conditions. The test statistic denominator is the combined pooled standard error for the two experimental conditions obtained by looking up the var.M from each baseOlig.error variance function. The conversion to p-values is based on the Gaussian distribution for difference if order statistics (medians). The user may select the smoother degrees of freedom (smaller is smoother) and the trim percent to obtain a variance function to suit particular issues i.e. variability of genes with low expression intensity. The default values for the adjustment of the variances of the two groups is the asymptotically correct value of \(\pi/2\). This value is biased at small sample values and unbiased adjustment parameters based on sample size can be used instead. See documentation of lpeAdj for details.

Value

Data frame including x, median of x, y, median of y, median difference of \((x,y)\), pooled standard deviation of difference, LPE p-value, outlier flag, probability of an outlier within x or y.
Author(s)

Carl Murie (carl.murie@mcgill.ca), Nitin Jain (nitin.jain@pfizer.com)

References


Jain et. al. (2003) *Local pooled error test for identifying differentially expressed genes with a small number of replicated microarrays*, Bioinformatics, 1945-1951.


See Also

lpeAdj

Examples

```r
# Loading the library and the data
library(LPEadj)
data(Ley)

ADJ.VALUES <- c(1, 1, 1.34585905516761, 1.1936328146169, 1.436849413109,
                 1.289652132873, 1.47658053092781, 1.34382984832146,
                 1.49972130857404, 1.3835405678718)

dim(Ley)
# Gives 124887
# First column is ID.

# Subsetting the data
subset.Ley <- Ley[1:1000,]
subset.Ley[,2:7] <- preprocess(subset.Ley[,2:7], data.type="MAS5")

# Finding the baseline distribution of condition 1 and 2.
var.1 <- adjBaseOlig.error(subset.Ley[,2:4], q=0.01, setMax1=FALSE)
var.2 <- adjBaseOlig.error(subset.Ley[,5:7], q=0.01, setMax1=FALSE)

# Applying LPE
lpe.result <- calculateLpeAdj(subset.Ley[,2:4], subset.Ley[,5:7], var.1, var.2,
                              probe.set.name=subset.Ley[,1], adjust1=ADJ.VALUES[3],
                              adjust2=ADJ.VALUES[3])
```
getCols

Get number of replicates from character vector

Description

Get number of replicates from character vector

Usage

getCols(labels)

Arguments

labels

character vector showing whether each column is control or treatment. It is assumed that the columns are ordered, the first n1 reps are control the next n2 reps are treatment

Value

Return vector of length two which are the number of reps for the first second group.

Author(s)

Carl Murie (carl.murie@mcgill.ca)

lpeAdj

High level lpeAdj function that executes the adjusted local pooled error significance test. If more control over parameters is needed then see documentation for calculateLpeAdj.

Description

Applies the LPE algorithm with two additional options. The first is that the original LPE method sets all variances below the max variance in the ordered distribution of variances to the maximum variance. in LPEadj this option is turned off by default. The second option is to use a variance adjustment based on sample size rather than pi/2. By default the LPEadj uses the sample size based variance adjustment. It is recommended to keep both of these options to the default.

Usage

lpeAdj(dat, labels=NULL, doMax=FALSE, doAdj=TRUE, q=.01)
Arguments

- **dat**: Replicated data of experiment containing two groups (as matrix or data-frame)
- **labels**: vector of group labels that correspond to the columns of dat. eg. labels=c(0,0,0,1,1,1) describes two groups with three replicates each
- **doMax**: boolean: if T then all variances below the max variance in the ordered distribution of variances are set to the maximum variance. It is recommended to use the default value of False.
- **doAdj**: If T then run LPE with using variance adjustment value based on number of replicates (hardcoded in adjValues) rather than pi/2.
- **q**: is the quantile width; q=0.01 corresponds to 100 quantiles i.e. percentiles. Bins/quantiles have equal number of genes and are split according to the average intensity A.

Details

The LPE test statistic numerator is the difference in medians between the two experimental conditions. The test statistic denominator is the combined pooled standard error for the two experimental conditions obtained by looking up the var.M from each baseOlig.error variance function. The conversion to p-values is based on the Gaussian distribution for difference if order statistics (medians).

Value

Data frame including x, median of x, y, median of y, median difference of (x,y), pooled standard deviation of difference, LPE p-value, outlier flag, probability of an outlier within x or y.

Author(s)

Carl Murie (carl.murie@mcgill.ca), Nitin Jain (nitin.jain@pfizer.com)

References


Examples

```r
# Loading the library and null dataset (two groups with three replicates each)
library(LPEadj)
dat <- matrix(rnorm(6000), ncol=6)

# Applying LPE
lpe.result <- lpeAdj(dat, labels=c(0,0,0,1,1,1))
```
Index

*Topic methods*
  - adjBaseOlig.error, 1
  - adjBaseOlig.error.step1, 2
  - adjBaseOlig.error.step2, 4
  - calculateLpeAdj, 6
  - getCols, 8
  - lpeAdj, 8

adjBaseOlig.error, 1
adjBaseOlig.error.step1, 2
adjBaseOlig.error.step2, 4

calculateLpeAdj, 6

getcols, 8

lpeAdj, 2, 3, 5, 7, 8